XCP-i : "eXplicit Control Protocol" pour l'interconnexion de réseaux haut-débit hétérogènes

Dino Martín Lopez Pacheco, Laurent Lefèvre, Cong-Duc Pham

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

XCP-i : “eXplicit Control Protocol” pour l’interconnexion de réseaux haut-débit hétérogènes

Dino M. Lopez Pacheco — Laurent Lefèvre — Congduc Pham

N° 6385

Décembre 2007

Thème NUM

ISSN 0249-6399 ISRN INRIA/RR--6385--FR+ENG

XCP-i : "eXplicit Control Protocol" pour l'interconnexion de réseaux haut-débit hétérogènes

Dino M. Lopez Pacheco [*] , Laurent Lefèvre [†] , Congduc Pham [‡]

Thème NUM — Systèmes numériques
Projet RESO

Rapport de recherche n° 6385 — Décembre 2007 — 16 pages

Résumé : *eXplicit Control Protocol* (XCP) est un protocole de transport qui contrôle efficacement l'évolution de la fenêtre de congestion de l'émetteur, évitant ainsi la phase de slow-start et d'évitement de congestion. XCP nécessite cependant la collaboration de tous les routeurs sur le chemin de la source vers le récepteur, ce qui est pratiquement impossible à réaliser en réalité, sinon ses performances peuvent être beaucoup moins bonnes que celles de TCP. Cette forte dépendance de XCP en des routeurs spécialisés limite considérablement l'intérêt de déployer des routeurs XCP sur une portion du réseau. Nous proposons dans cet article une extension de XCP, appelée XCP-i, qui permet d'interconnecter des nuages XCP avec des nuages non-XCP sans perdre le bénéfice du contrôle précis de XCP sur la fenêtre de congestion. Les résultats de simulation sur des topologies correspondant typiquement à des scénarios de déploiement incrémental montrent que les performances de XCP-i sont bien supérieures à celles de TCP sur des liens à haut-débit.

Mots-clés : Réseaux haut-débit hétérogènes, nuage non-XCP, routeur virtuel XCP

This text is also available as a research report of the Laboratoire de l'Informatique du Parallélisme `http://www.ens-lyon.fr/LIP`.

[*] CONACyT - INRIA RESO / Université de Lyon - LIP (UMR 5668 CNRS, ENS, INRIA, UCBL), France dmlopezp@ens-lyon.fr

[†] INRIA RESO / Université de Lyon - LIP (UMR 5668 CNRS, ENS, INRIA, UCBL), France laurent.lefevre@inria.fr

[‡] LIUPPA, Université de Pau, France Congduc.Pham@univ-pau.fr

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier (France)
Téléphone : +33 4 76 61 52 00 —Télécopie +33 4 76 61 52 52

XCP-i : eXplicit Control Protocol for the interconnection of heterogeneous high speed networks

Abstract: *eXplicit Control Protocol* (XCP) is a transport protocol that efficiently controls the sender's congestion window size thus avoiding the slow-start and congestion avoidance phase. XCP requires the collaboration of all the routers on the data path which is almost impossible to achieve in an incremental deployment scenario of XCP. If not, XCP's performances are worse than those of TCP thus limiting dramatically the benefit of having XCP running in some parts of the network. In this paper, we address this problem and propose XCP-i which is operable on an internetwork consisting of XCP routers and traditional IP routers without loosing the benefit of the XCP control laws which allow the congestion window to jump directly to the optimal size. The simulation results on a number of topologies that reflect the various scenario of incremental deployment on the Internet show that XCP-i outperforms TCP on high-speed links.

Key-words: High speed heterogeneous networks, non-XCP cloud, virtual XCP router

1 Introduction

Dans les réseaux haut-débit où la capacité des liens peut être de l'ordre de plusieurs Gbits/s, le protocole de transport TCP doit être optimisé pour pouvoir profiter du haut-débit : augmentation de la taille du *socket buffer*, augmentation de la valeur maximale pour la fenêtre de congestion... Néanmoins, ces réglages donnent des performances limitées ; ce qui pénalise le plus TCP c'est la lenteur de la phase d'évitement de congestion où la fenêtre de congestion n'augmente que d'un paquet par aller-retour. C'est pour tenter de résoudre ce problème que différentes solutions ont été proposées. Par exemple, HSTCP [11] modifie la fonction de réponse de TCP pour récupérer plus rapidement le débit disponible mais aussi pour limiter les baisses en débit faisant suite aux pertes de paquets. Ce comportement plus agressif affecte l'équité intra et inter protocole. FAST TCP [2] est une amélioration de TCP Vegas qui utilise les variations du RTT pour anticiper les congestions. FAST TCP présente de bonnes performances mais ne permet pas de prendre en compte les cas où l'augmentation du RTT n'est pas liée à une congestion comme c'est le cas dans les opérations de re-routage par exemple. Alors que TCP, HSTCP et FAST TCP sont des approches de bout-en-bout, XCP [5] appartient à la catégorie des protocoles avec assistance des routeurs. XCP utilise des routeurs spécialisés qui permettent de signaler et d'informer de manière très précise sur l'état de congestion dans le réseau permettant ainsi à la source de déterminer la taille optimale de sa fenêtre de congestion et maximiser de cette façon l'utilisation des liens ainsi que le niveau d'équité.

XCP est donc une solution très prometteuse. Plusieurs études ont montré analytiquement ses performances [8], ont proposé des améliorations pour le rendre plus robuste [7, 6] ou encore ont réalisé des études expérimentales sur des implémentations réelles [12]. Dans la plupart de ces études, le problème du déploiement incrémental a été abordé et il a été montré que la présence de routeurs non-XCP entre la source et le récepteur dégradait très fortement les performances de XCP. Cette forte dépendance de XCP envers des routeurs spécialisés limite considérablement l'intérêt de déployer des routeurs et des clients XCP. Dans [5] les auteurs proposent un modèle de déploiement incrémental basé sur la création de nuages XCP avec des routeurs de bordure qui traduisent les flux non-XCP vers XCP. Cependant cette idée très complexe n'a pas été développée ni testée. Nous proposons donc dans cet article une extension de XCP, appelée XCP-i (XCP inter-opérable), qui permet d'interconnecter des routeurs XCP et des routeurs non-XCP. XCP-i permet de conserver intact les couches de contrôle de XCP. Les nouvelles fonctionnalités apportées par XCP-i n'augmentent que légèrement la complexité du protocole XCP. Les résultats de simu-

lation sur des topologies correspondant typiquement à des scénarios de déploiement incrémental montrent que nos extensions sont efficaces.

L'article est organisé de la manière suivante : la section 2 rappelle le fonctionnement de base de XCP. La section 3 présente les objectifs de XCP-i et les mécanismes que nous proposons pour détecter les nuages non-XCP et prendre en compte les ressources disponibles à l'intérieur de ces nuages. La section 4 présente les résultats de simulation. Nous discuterons de quelques limitations dans la section 5 et conclurons notre article par la section 6.

2 Le protocole XCP

2.1 Description générale

XCP [5] (*eXplicit Control Protocol*) est un protocole qui utilise l'assistance des routeurs pour informer précisément l'émetteur des conditions de congestion du réseau. Les paquets de données XCP comportent un en-tête de congestion, rempli par l'émetteur, qui contient la taille actuelle de la fenêtre de congestion de celui-ci (le champ `H_cwnd`), l'estimation du temps aller-retour (le champ `rtt`) et une valeur appelée $feedback$ (le champ `H_feedback`), qui indique à l'émetteur un incrément (si elle est positive) ou un décrément (si elle est négative) à appliquer à sa fenêtre de congestion. Le champ `H_feedback` est le seul qui peut être modifié par les routeurs XCP en fonction des valeurs des 2 autres champs. Quand le récepteur reçoit un paquet de données, il recopie l'en-tête du paquet dans l'en-tête d'un paquet d'accusé de réception (ACK) qui sera envoyé vers l'émetteur.

À la réception de l'ACK, l'émetteur mettra à jour la taille de sa fenêtre de congestion de la manière suivante : $cwnd = max(cwnd + H\_feedback, packetsize)$, où $cwnd$ est exprimé en octets. Le mécanisme central d'un routeur XCP est basé sur l'utilisation d'un contrôleur d'efficacité (EC) et d'un contrôleur d'équité (FC) qui réalisent la mise à jour de $feedback$ pendant un intervalle de contrôle équivalent à la moyenne des RTT. L'EC a la responsabilité de maximiser l'utilisation de la bande passante tout en minimisant le nombre de paquets rejetés et le FC de partager les ressources de façon équitable. Il va assigner à $feedback$ une valeur proportionnelle à la bande passante disponible (S), déduite de la différence entre le trafic entrant total et la capacité du lien de sortie. La taille résiduelle Q de la file d'attente est également prise en compte. Dans une deuxième étape, le FC traduit cette valeur $feedback$ globale (qui peut être assimilée à une valeur agrégée de bande passante disponible positive ou négative) en une valeur $feedback$ par paquet (qui sera ensuite placée dans l'en-tête de chaque paquet de données) en suivant des règles d'équité

par flux similaires aux règles AIMD de TCP. Il faut noter qu'il n'y a pas d'états par flux conservés par le routeur XCP pour exécuter toutes ces opérations. En effet, comme les paquet de données d'un flux donné portent dans leur en-tête la valeur actuelle de la fenêtre de congestion et le RTT, il est possible de calculer pour chaque flux le nombre de paquets envoyés par fenêtre de congestion afin d'assigner la bande passante disponible de manière proportionnelle.

XCP est donc capable d'atteindre très rapidement le débit optimal et de réagir aussi rapidement aux variations de bande passante. Cependant, dans le cas de cohabitation avec TCP, XCP récupérera moins de bande passante à cause du contrôle strict qu'il impose sur le taux de pertes.

2.2 Fragilité de XCP avec des routeurs non-XCP

Comme XCP repose sur des routeurs spécialisés pour estimer la bande passante disponible tout le long du chemin, de l'émetteur jusqu'au récepteur, il est très probable que XCP réagisse mal en présence de routeurs non-XCP (le terme **routeur non-XCP** fait référence à un routeur IP traditionnel, par exemple : Drop Tail, RED, etc. Un **nuage non-XCP** est un ensemble de n routeurs non-XCP où $n > 1$). Dans ce cas nous pouvons aussi prédire que XCP sera beaucoup moins performant que TCP car le $feedback$ calculé ne prendra en compte que les éléments XCP sur le chemin ignorant ainsi l'existence d'un éventuel goulot d'étranglement.

Ce comportement a été démontré dans [12] et nous incluons ici des simulations qui montrent ces problèmes pour rendre notre article plus clair. La figure 1 présente 3 scénarios : (a) montre un réseau Internet typique avec des routeur non-XCP, (b) montre un réseau 100% XCP et (c) montre un scénario de déploiement incrémental de XCP autour d'un routeur non-XCP.

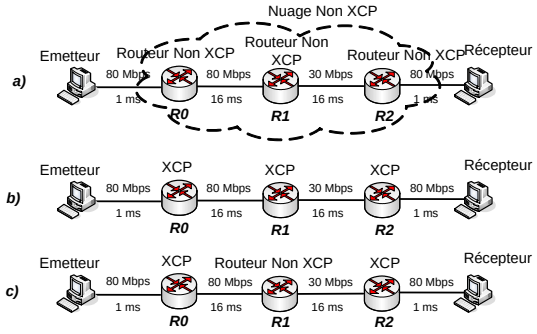

FIG. 1 – (a) scénario pour TCP, (b) et (c) scénarios pour XCP.

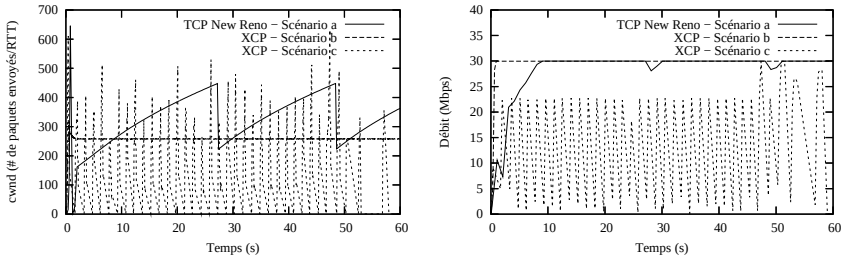

FIG. 2 – Évolution de la fenêtre de congestion et débit pour les scénarios a,b et c.

La figure 2 montre le comportement d'un unique flux TCP sur le scénario (a) et d'un unique flux XCP sur les scénarios (b) et (c). L'évolution de la fenêtre de congestion (partie gauche de la figure) montre la courbe en dent de scie typique de TCP et celle de XCP qui atteint rapidement la valeur optimale en évitant toujours la perte de paquets. Pour XCP sur le scénario (c), la fenêtre de congestion est très instable et dépasse très souvent celle de TCP. En effet, étant donné que les routeurs non-XCP ne sont pas capable de faire la mis à jour du $feedback$ portée dans l'en-tête du paquet XCP pour indiquer la capacité du goulot d'étranglement, le prochain routeur XCP utilisera une valeur qui reflète la bande passante disponible avant le nuage non-XCP, laquelle est beaucoup plus grande que celle du goulot d'étranglement à 30 Mbits/s que nous avons utilisé dans notre scénario. Dans ces simulation faites sur le simulateur *ns*, TCP sur le scénario (a) a envoyé 215.004 Mo, XCP sur le scénario (b) a envoyé 223.808 Mo, et XCP sur le scénario (c) a seulement envoyé 52.426 Mo pendant une minute !

3 Extension de XCP pour l'interconnexion de réseaux hétérogènes

La section précédente a montré les problèmes de XCP à gérer correctement les nuages non-XCP. Cette section décrit le mécanisme que nous proposons pour rendre XCP performant dans un scénario de déploiement incrémental. Nous appellerons XCP-i cette nouvelle version de XCP (le caractère *i* dans ce cas indique le terme inter-opérable). Nous utiliserons alors le terme **routeur XCP-i** pour faire référence à un routeur XCP avec des fonctionnalités d'*inter-opérabilité*. Rendre XCP inter-opérable doit cependant se faire avec le minimum de changements et plus particulièrement en gardant le mécanisme central et la philosophie de XCP qui consiste à éviter

de conserver des états par flux. Une des principales raisons à cela est qu'il existe actuellement des implémentations de XCP qui ont montré que cette implémentation n'est pas triviale [12].

L'extension XCP-i ajoute deux nouvelles fonctionnalités : (*i*) la détection des paquets qui ont traversés un nuage non-XCP et (*ii*) la prise en compte de la bande passante disponible dans le nuage non-XCP dans les calculs du $feedback$. Nous présenterons dans les sections suivantes comment nous avons implémenté ces nouvelles fonctionnalités, toujours en conservant les mécanismes centraux de XCP inchangés.

3.1 XCP-i : architecture et algorithmes dans les routeurs

3.1.1 Détection d'un nuage non-XCP

XCP-i détecte les nuages non-XCP grâce au compteur TTL. Nous supposons que tous les routeurs du réseau sont capables de décrémenter la valeur du TTL dans l'en-tête du paquet IP avant d'envoyer le paquet sur le lien de sortie. Avec cette hypothèse, nous ajoutons un nouveau champ dans l'en-tête du paquet XCP appelé `xcp_ttl_`, qui est décrémenté uniquement par les routeurs XCP-i. Les champs TTL et `xcp_ttl_` doivent être initialisés avec la même valeur par l'émetteur. Dans un réseau complètement XCP, les champs TTL et `xcp_ttl_` auront ainsi toujours la même valeur. Par contre, quand un routeur XCP-i reçoit un paquet avec un champ TTL plus petit que `xcp_ttl_`, il peut conclure que ce paquet vient de traverser un nuage non-XCP. Cette solution est simple, n'a besoin d'aucun échange de messages spécifiques entre les routeurs, et le temps de traitement de ce nouveau champ reste négligeable.

3.1.2 Détection des routeurs XCP de bordure

Quand un nuage non-XCP a été détecté par un routeur XCP-i, XCP-i a besoin de connaître l'identité du dernier routeur XCP-i avant le nuage non-XCP. En effet, XCP-i va essayer de déterminer la bande passante disponible entre les 2 routeurs XCP-i placés autour du nuage non-XCP. Pour pouvoir découvrir l'identité du routeur émetteur nous avons ajouté un nouveau champ dans l'en-tête du paquet XCP, appelé `last_xcp_router_`, qui contient l'adresse IP du dernier routeur XCP-i qui a traité le paquet. Un routeur XCP-i devra simplement mettre son adresse IP dans ce champ avant d'envoyer le paquet sur le lien de sortie. De cette manière, quand un nuage non-XCP est détecté par un routeur XCP-i, celui-ci connaîtra automatiquement quel est le dernier routeur XCP-i placé de l'autre côté du nuage non-XCP. Encore une

fois, cette solution est simple, n'a besoin d'aucun échange de messages spécifiques entre les routeur XCP-i et le temps de traitement d'un paquet reste minimal.

3.1.3 Détermination de la bande passante disponible dans le nuage non-XCP

Tout d'abord, nous appellerons XCP-i$_{k-1}$ et XCP-i$_k$ les 2 routeurs autour du nuage non-XCP. L'idée principale derrière XCP-i est de démarrer une procédure d'estimation de bande passante dans le routeur XCP-i$_{k-1}$. Pour cela XCP-i$_k$ envoie une requête à XCP-i$_{k-1}$ et attend un accusé de réception pendant une période *xcp_req_timeout*. Si cet accusé de réception n'arrive pas à temps le processus est redémarré. Après 3 requêtes infructueuses, XCP-i$_k$ conclut que le chemin entre XCP-i$_{k-1}$ et lui est rompu ou n'existe pas. La procédure d'estimation de bande passante sera re-exécutée après la réception d'un nouveau paquet de données envoyé par XCP-i$_{k-1}$. Maintenant, si la requête envoyée a bien été reçue par XCP-i$_{k-1}$, celui-ci devra envoyer un accusé de réception à XCP-i$_k$ et essaiera de déterminer la bande passante disponible entre XCP-i$_{k-1}$ et XCP-i$_k$. Plusieurs algorithmes pour déterminer la bande passante disponible ont été proposés dans la littérature scientifique (par exemple *packet pair*, *packet train*, etc.). Nous allons supposer que le routeur exécutera un de ces algorithmes et qu'il trouvera l'estimation la plus proche possible de la valeur réelle (par exemple les auteurs dans [3] ont montré que *pathchirp* [10], *pathload* [4] ou *Iperf* [9] donnent des estimations très proches de la valeur réelle). Une fois que la bande passante disponible, notée $BW_{k-1,k}$, sera obtenue elle devra être envoyée à XCP-i$_k$ qui créera une entrée dans une table de hachage, en utilisant l'adresse IP de XCP-i$_{k-1}$, pour y conserver la bande passante disponible entre XCP-i$_{k-1}$ et XCP-i$_k$ (soit $BW_{k-1,k}$). La procédure d'estimation de la bande passante disponible devra être exécutée périodiquement par XCP-i$_{k-1}$. Cette procédure devra être arrêtée après une période d'inactivité de XCP-i$_{k-1}$ et l'entrée correspondante dans la table de hachage supprimée.

Il est important que ce soit XCP-i$_k$ qui stocke la bande passante disponible (et exécute donc les mécanismes XCP pour calculer le $feedback$ comme il sera expliqué dans la prochaine section) et non XCP-i$_{k-1}$ car celui-ci est incapable de différencier les paquets qui arriveront à XCP-i$_k$ de ceux qui arriveront à un autre routeur XCP-i après d'avoir traversé le nuage non-XCP (voir figure 3 pour un exemple). C'est la raison pour laquelle XCP-i$_{k-1}$ doit communiquer la bande passante disponible à XCP-i$_k$. Cette solution ne conserve pas d'état par flux mais un état par routeur XCP-i de bordure, soit un nombre assez faible d'états.

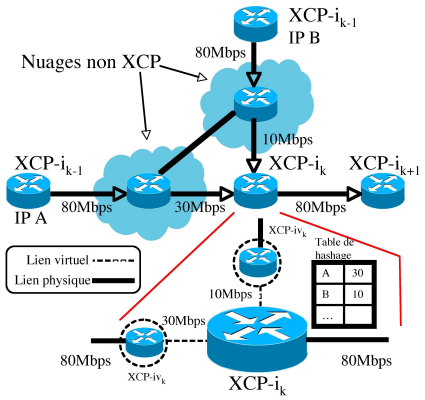

FIG. 3 – Routeur XCP-i avec un routeur virtuel par nuage non-XCP.

3.1.4 Le routeur virtuel XCP-iv

Quand XCP-i$_k$ reçoit un paquet qui a traversé un nuage non-XCP, il vérifie si une entrée $BW_{k-1,k}$ existe dans la table de hachage pour `last_xcp_router_`. Si c'est le cas, XCP-i$_k$ utilisera un routeur virtuel, XCP-iv$_k$, pour calculer le $feedback$ qui reflétera les conditions du réseau dans le nuage non-XCP. L'objectif du routeur virtuel est de simuler un routeur XCP-i placé juste avant XCP-i$_k$, avec un lien virtuel de sortie relié à ce dernier routeur et une capacité égale à la bande passante disponible trouvée dans le nuage non-XCP. La figure 3 montre la structure logique du routeur XCP-i$_k$ avec un routeur virtuel par nuage non-XCP.

Nous pouvons considérer le routeur virtuel comme une entité logique qui remplace le nuage non-XCP. L'équation pour calculer le $feedback$ dans XCP-iv est alors similaire à l'équation utilisée par XCP (nous pouvons donc réutiliser le code XCP original) :

$$feedback_{XCP-iv_k} = \alpha.rtt.BW_{k-1,k} - \beta.Q \quad (1)$$

Les valeurs de α et β sont celles de l'algorithme XCP. rtt et Q sont respectivement la moyenne du RTT et la taille résiduelle de la file d'attente du routeur XCP-i qui contient le routeur virtuel. Dans l'équation (1) $BW_{k-1,k}$ remplace donc S dans l'équation originale de XCP. De cette manière le routeur virtuel n'a pas besoin de connaître le trafic entrant (voir la section 2.1) car une fois que $feedback$ est mis à jour par le routeur virtuel, XCP-i$_k$ exécutera tous les calculs nécessaires pour trouver son propre $feedback$ comme le ferait un routeur XCP normal.

3.2 XCP-i : architecture dans les noeuds terminaux

Il est possible que lors d'un déploiement incrémental de XCP, l'émetteur ou le récepteur, ou bien les deux, ne se trouvent pas connectés directement à un routeur XCP. Par exemple, la figure 4 montre un scénario où les routeurs XCP-i ont été placés du côté du récepteur, l'émetteur étant connecté à un nuage non-XCP.

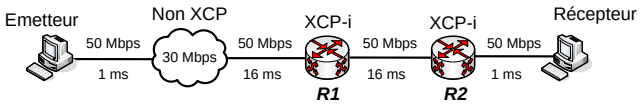

FIG. 4 – Déploiement asymétrique : Optimisation du côté du récepteur

Dans ce cas, des parties de l'algorithme XCP-i devront aussi être supportées par les noeuds terminaux. Si le routeur XCP-i est placé du côté du récepteur (figure 4), l'émetteur devra être capable d'initialiser une procédure d'estimation de bande passante à partir de la réception d'une requête de la part du premier routeur XCP-i sur le chemin. Quand le routeur XCP-i est placé du côté de l'émetteur, le récepteur devra agir comme un routeur XCP-i en exécutant la procédure de détection des nuages non-XCP et le calcul d'un $feedback$ correspondant à la valeur de la bande passante disponible trouvée dans les nuages non-XCP. Si cette solution n'est pas souhaitable, le récepteur devra demander au dernier routeur XCP-i de calculer le $feedback$ à sa place. Nous pensons que cette dernière solution est bien plus complexe que la première, qui possède l'avantage de simplement dupliquer le code de XCP-i dans la pile protocolaire XCP du récepteur (voir section 3.1.4).

4 Résultats de simulation

Notre modèle de XCP-i a été développé sur une extension du modèle *ns* de XCP de Katabi. Si non précisé, nous supposerons que l'estimation de bande passante disponible renvoie la valeur correcte à la fin de chaque intervalle de contrôle XCP.

4.1 Scénario de Déploiement Incrémental autour de nuages non-XCP

Le premier scénario sur lequel XCP-i a été expérimenté concerne un déploiement symétrique dans des points de *peering* du réseau (figure 5) où 2 nuages non-XCP sont connectés par un routeur XCP-i. Les résultats de simulation montrent que la fenêtre

de congestion de l'émetteur et le débit du récepteur sont stables (Figure 6) avec des s similaires au scénario où tous les routeurs sont XCP (partie *b* de la figure 1). De plus, nous n'observons aucune perte de paquets. Le routeur XCP-i virtuel dans R1 et R2 estime la bande passante disponible dans le nuage non-XCP et calcule ainsi la valeur de $feedback$ optimale. Ces résultats montrent que XCP-i est capable de supporter efficacement des flux hautes performances dans des réseaux hétérogènes même si son déploiement se limite à quelques endroits stratégiques du réseau.

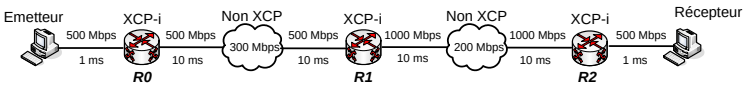

FIG. 5 – Déploiement incrémental sur des points de *peering*

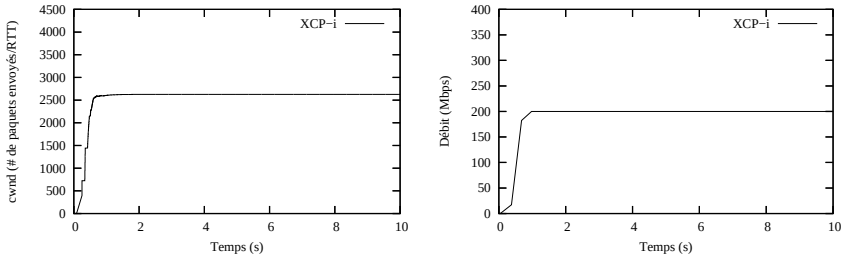

FIG. 6 – Fenêtre de congestion et débit dans le scénario de déploiement incrémental

4.2 Scénario de Fusion : n nuages non-XCP partageant un chemin XCP

Le scénario de fusion repose sur une topologie où 2 nuages non-XCP partagent un chemin XCP (Figure 7). Nous validons ainsi la capacité du protocole XCP à garantir l'équité entre 2 flux agrégés. Le routeur XCP-i R1 doit générer un routeur virtuel XCP-iv pour chaque lien connecté à un nuage non-XCP. La figure 8 démontre que XCP-i réussit à maintenir une équité des flux avec les émetteurs i et j qui obtiennent respectivement 280Mbits/s et 100Mbits/s.

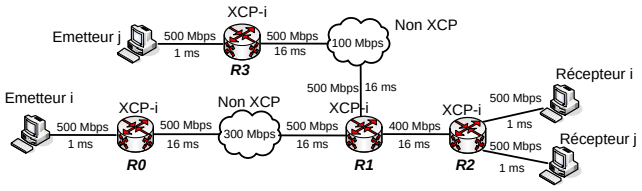

FIG. 7 – 2 files d'attente non-XCP partageant un chemin XCP

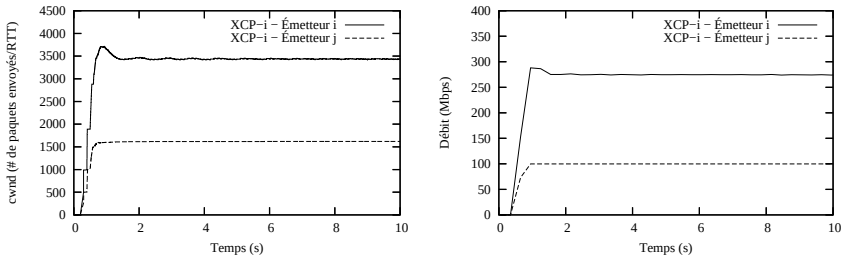

FIG. 8 – Fenêtre de congestion et débit dans le scénario de Fusion

4.3 Scénario de Duplication : un nuage non-XCP dessert n chemins XCP

La figure 9 décrit une topologie où un nuage non-XCP est connecté à 2 chemins XCP. Les résultats de la figure 10 démontrent que XCP-i est capable de partager équitablement le lien de 500Mbits/s en 2 flux de 250Mbits/s.

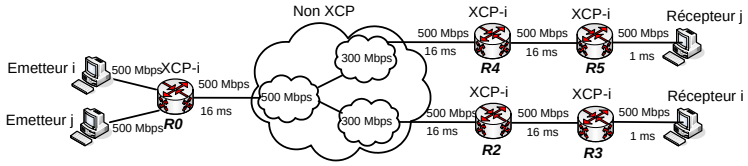

FIG. 9 – Une file d'attente non-XCP est partagée par des noeuds XCP

4.4 Variation de la précision de l'estimation de bande passante

Nous avons supposé jusqu'à présent que l'estimation de bande passante disponible était parfaite. Ce n'est pas toujours le cas [3] et dans certaines conditions, les outils utilisés peuvent sous ou surestimer la bande passante disponible. Nous avons

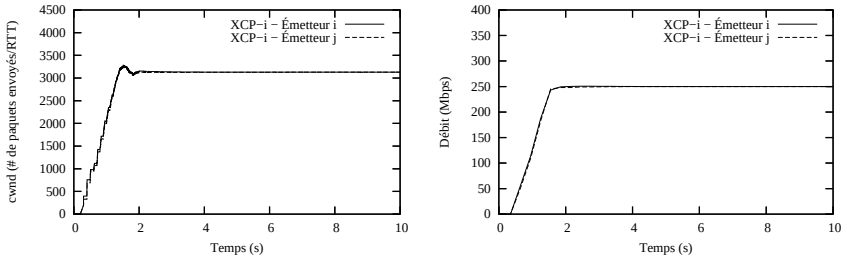

FIG. 10 – Fenêtre de congestion et débit dans le scénario de duplication

mené une série de simulations basées sur la topologie de la figure 7 afin de comparer XCP-i et TCP sur des liens hautes performances lorsque l'estimation de bande passante est faussée (sous et sur-estimation de 10% et 20%).

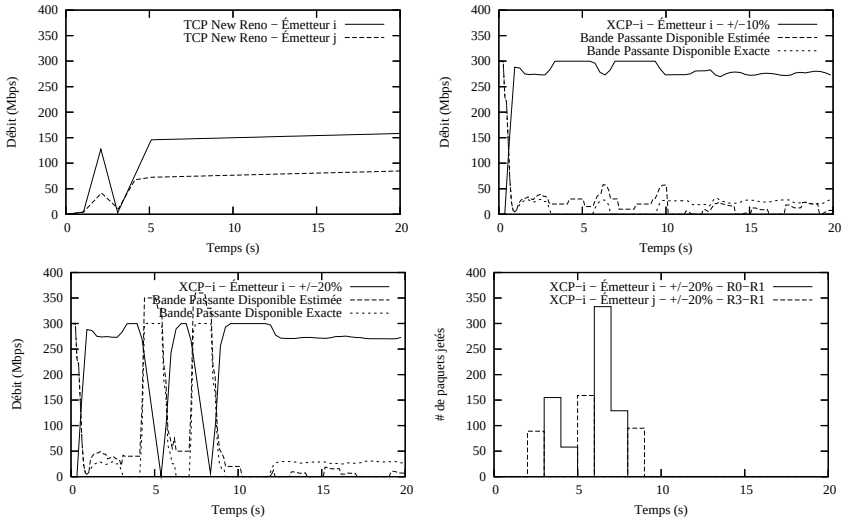

FIG. 11 – (h.g.) débit pour TCP. (h.d.) XCP-i, Emetteur i, 10%. (b.g) XCP-i, Emetteur i, 20%. (b.d.) pertes de paquets XCP-i, 20%

La figure 11 présente les débits pour les émetteurs i et j, la bande passante réelle et celle estimée. En haut à gauche, nous observons que TCP est incapable d'utiliser toute la bande passante disponible (capacités des liens à 300 et 100Mbits/s) et les émetteurs i et j envoient respectivement 329Mo et 172Mo en 20s. Avec une erreur d'estimation de 10% ou 20%, XCP-i continue de fonctionner efficacement : les émet-

teurs i et j envoient respectivement 690Mo et 182Mo avec 10% d'erreur et 590Mo et 187Mo avec 20% d'erreur (XCP-i avec estimation parfaite permet à i et j de transférer 670Mo et 244Mo). Le principal problème causé par la sur-estimation de bande passante concerne la perte de paquets et les *timeouts*. Bien que les performances de XCP-i dépendent de la précision de l'estimation de bande passante disponible, les débits des flux utilisant XCP-i dépassent ceux des flux TCP car XCP-i rétablit très rapidement son débit après des pertes de paquets.

5 Problèmes restant à résoudre

5.1 Équité et sur-estimation dans un nuage non-XCP

La figure 12 montre une topologie qui n'est pas totalement supportée par XCP-i : lorsqu'un goulot d'étranglement dans un nuage non-XCP est partagé par 2 chemins XCP. Dans le cas général, lorsque plus de 2 routeurs XCP-i de bordure partagent le même goulot d'étranglement ceux-ci vont sur-estimer la bande passante totale disponible. Tous les autres cas sont exclus du problème décrit dans cette partie. Dans la figure 12 chaque couple de routeurs (a,b) et (d,c) va indépendamment trouver la bande passante disponible et autoriser respectivement les émetteurs i et j à transmettre à 300Mbit/s ce qui va créer une charge de 600Mbit/s sur le goulot d'étranglement. Un second problème de cette topologie est celui de l'équité lorsqu'il y a déjà un flux XCP qui prend pratiquement toute la capacité du lien. Quand un deuxième émetteur va vouloir émettre, XCP-i ne sera pas capable d'allouer de manière équitable la bande passante car ce sont 2 chemins XCP distincts. Le deuxième flux va donc uniquement récupérer la bande passante qui sera réservée par le mécanisme du *bandwidth shuffling* de XCP, soit 10% au maximum (voir [5]). Ces 2 problèmes seront étudiés dans nos travaux futurs.

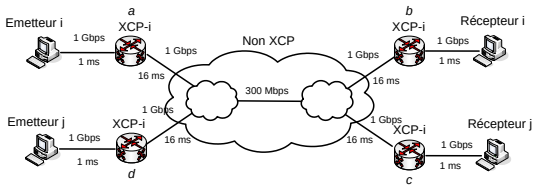

FIG. 12 – Un goulot d'étranglement partagé par n chemins XCP

5.2 Équité avec TCP

Le problème d'équité avec TCP n'est pas un problème de XCP-i mais celui de XCP en général. En effet, XCP n'est capable que de récupérer la bande passante disponible avec l'objectif de ne pas causer de pertes de paquets. Il est donc très peu, voire pas du tout, agressif. XCP-i étant basé sur le contrôle de XCP, il présente le même problème. Dans cet article, nous n'avons pas abordé le problème de l'équité avec TCP : les nuages non-XCP peuvent transporter des flux non-XCP mais XCP-i ne pourra récupérer que la bande passante disponible laissée par ces flux non-XCP. Le problème de la cohabitation de XCP et de TCP sera pris en compte dans nos travaux futurs.

6 Conclusion et travaux futurs

Nous avons présenté dans cet article une extension de XCP, appelée XCP-i, qui permet d'utiliser le protocole XCP pour l'interconnexion de réseaux IP haut-débit. La principale ligne directrice de XCP-i est de garder le processus de contrôle de XCP inchangé tout en ajoutant de nouvelles fonctionnalités pour la détection et la gestion des nuages réseaux non XCP. Les résultats de simulation montrent que, sur un large éventail de cas, XCP-i réussit à fournir un niveau de performance égal à celui de XCP. Même si les performances de XCP-i peuvent dépendre de la précision de l'estimation de la bande passante disponible dans un nuage non-XCP, celles-ci sont bien supérieures à celles de TCP sur des liens à haute capacité. XCP-i, tout comme XCP, récupère très rapidement le débit disponible après des pertes de paquets. Nos travaux actuels concernent l'implémentation de XCP-i sur des routeurs programmables. Cela nous permettra de mener des expérimentations à grande échelle de XCP-i sur la plate-forme Grid5000 [1].

Remerciements

Une partie de ces travaux est supportée par le Consejo Nacional de Ciencia y Tecnología (CONACyT - www.conacyt.mx).

Références

[1] F. Cappello, F. Desprez, M. Dayde, E. Jeannot, Y. Jegou, S. Lanteri, N. Melab, R. Namyst, P. Primet, O. Richard, E. Caron, J. Leduc, and G. Mor-

net. Grid'5000 : A large scale, reconfigurable, controlable and monitorable grid platform. In *6th IEEE/ACM International Workshop on Grid Computing, Grid'2005*, Seattle, Washington, USA, November 2005.

[2] David X. Wei Cheng Jin and Steven H. Low. FAST TCP : Motivation, Architecture, Algortihms, Performance. In *INFOCOM*. IEEE, March 2004.

[3] Alok Shriram et al. Comparison of Public End-to-End Bandwidth Estimation Tools on High-Speed Links. In *PAM*, 2005.

[4] M. Jain and C. Dovrolis. Pathload : An Available Bandwidth Estimation Tool. In *PAM*, 2002.

[5] D. Katabi, M. Handley, and C. Rohrs. Congestion control for high bandwidth-delay product networks. In *ACM SIGCOMM*, 2002.

[6] Dino M Lopez-Pacheco and Congduc Pham. Robust Transport Protocol for Dynamic High-Speed Networks : Enhancing the XCP Approach. In *Proceedings of IEEE International Conference on Networks*, volume 1, pages 404–409, Kuala Lumpur, Malaysia, November 2005.

[7] Dino M Lopez-Pacheco and Congduc Pham. Enabling Large Data Transfers on Dynamic, Very High-Speed Network Infrastructures. In *Proceedings of ICN/ICONS/MCL 2006*, Mauritius, April 2006.

[8] S. H. Low, L. Andrew, and B. Wydrowsk. Understanding XCP : Equilibrium and Fairness. In *IEEE Infocom*, 2005.

[9] NLANR. Iperf v1.7.0. In *http ://dast.nlanr.net/projects/iperf*, 2004.

[10] V. Ribeiro. PathChirp : Efficient Available Bandwidth Estimation for Network Path. In *PAM*, 2003.

[11] S. Floyd. HighSpeed TCP for Large Congestion Windows. RFC 3649 (Experimental), December 2003.

[12] Y. Zhang and T. R. Henderson. An Implementation and Experimental Study of the eXplicit Control Protocol (XCP). In *INFOCOM*, pages 1037–1048, 2005.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399